Fall 2010

CS 209: Computer Programming for Business II

David M. Hutchison
Wright State University - Main Campus, david.hutchison@wright.edu

Syllabus

Course Information

Course title:	Computer Programming for Business II
Course number:	[redacted]n 01, Lab Section 01
Course discipline:	Computing Sciences
Course description:	CS 209 is the second in a sequence of two programming classes required for MIS majors. This course will continue teaching students to the basic concepts of programming. Examples are from business applications and emphasis is on problem solving with the computer as a tool.
Course date:	Monday, September 13, 2010 through Monday, November 15, 2010
Location:	355 Russ Engineering Center
Meeting day(s):	Monday
Meeting time(s):	6:05 - 9:35 (includes lecture and lab)
Prerequisite(s):	CS 208 or equivalent.

Instructor Information

Name:	David M. Hutchison
Email:	david.hutchison@wright.edu
Office location:	Inside Russ lounge
Office hours:	5:30 - 6:00 Monday
Biography:	• B.S. Computer Science, Wright State University, 1990 • M.B.A. Project Management, Wright State University, 1995 • Sun certified Java Programmer • Sun certified Java Developer • Project Management Professional • Current (Winter 2008-Present) instructor of CS 208 and CS 209 • Previous (1997-2001) and current instructor of CS 208 and CS 209 • Instructor of various independent Java, C programming, and Oracle-related Java courses
Teaching assistants:	Pavan Kapanipathi

Course Goal

Course goals:	This course, in conjunction with CS 208, is designed to help students achieve a high degree of proficiency in intermediate level programming skills.

Course Policies

Introduction:	All course policies are subject to change.

Course Requirements

Requirements:	This course assumes successful (i.e., passing) completion of CS 208 or equivalent. You are required to have a thumb drive or similar media. Although not required, a backup is highly recommended as well.

Course Schedule

Schedule: The course schedule presented in the WebCT Calendar is *tentative*. We will do our best to adhere to this schedule, but events beyond our control (e.g., weather) may impact the dates and content. I will do my best to not slip any exam dates.

Class Attendance

Policy: It is your responsibility to attend all classes - attendance will not be taken. Since this class is held only once a week, you stand to miss out on a lot of information, quizzes, labs, project hints, etc. if you choose to skip a class. Of course, prior notice of your missing class would be appreciated. Another way to think about it - if I haven't a clue of who you are because of your choosing not to attend class, then don't expect any leniency when it comes to "curving" grades.

Grading Policy

Policy: Grading is on a straight 10-point increment scale. That is, >90 is an A, 80-89 is a B, 70-79 is a C, 60-69 is a D, and <60 is an F. Weights of exams, assignments, etc. are as follows:

- 25% - Mid-term exam
- 25% - Final exam
- 30% - Programming assignments
- 10% - In-class labs
- 10% - Homework

Remember, your grade is weighted - it is *not* a straight points-based computation.

Academic Dishonesty

Policy: Violators will receive an F for the course and will be reported to the university - official university policy will be followed (Click here for the policy). You are required to work individually on your programming assignments. You are permitted to exchange ideas with your peers, but you are not permitted to use someone else's work. Additionally, you may not share your work with someone else. If you choose to violate these rules, then all students involved will suffer the consequences.

Course Lab

Lab: You must be enrolled in the lab associated with this class (i.e., CS 209 lab, section 01). The WebCT site for the lab will *not be used*.

In-class Lab Assignments

Lab assignments will be provided during each lab session. These assignments are to be worked on during lab and submitted prior to the end of the night's lab session, regardless of whether or not they are complete. These assignments will consist of straightforward coding problems such as writing source code for incomplete programs, or designing a complete Java application. There will be five lab assignments. Each lab will be worth 50 points.

Course Exams

Exams: All students are required to take both exams. Make-up exams are only given on a case-by-case basis. If you are unable to attend an exam, you are required to provide an acceptable and documented reason *prior* to the exam.

Course Quizzes

Quizzes: None

Programming Assignments

Assignments: There will be three programming assignments over the course of the quarter. Each of these assignments is worth 100 points, each will state the required due date, and each will state the requirements for that assignment (e.g., provide a design, test cases, source code, etc.). You are required to earn at least 60% of the total points for each *individual* assignment. Failure to earn 60% on each assignment with result in *failure of this course!* Failure to reach an average of 75% across all assignments will also result in *failure of this course!* Late assignments will only be accepted for documented reasons, previously arranged with the instructor (i.e., *not* the lab TA). Please see the lab TA for a description of how points will be allocated for each programming assignment.

Textbook

Required reading: *Java Programming: From the Ground Up*, Ralph Bravaco, Shai Simonson, McGraw-Hill, 1 edition (January 22, 2009), 978-0073523354

Midterm Exam

Content: Chapters 6 - 7.

Final Exam

Date: November 15, 2010 - 8:00pm til 10:00pm

Content: Comprehensive, with a concentration on chapters 9, 10, 12. I will do my best to schedule a review session from 7:00 til 8:00 that evening (location to be announced). You are welcome to ask any questions during the review session. My intent of that review session is to allow you to ask any last minute questions about the material on the final exam. This time is for you - use it!

Chapter 6

Lesson: Methods

Topics:

- Predefined methods
- Writing your own methods
- Local variables
- Scope and lifetime of variables
- Void methods
- Returning values from a method
- Overloading methods
- Passing parameters

Readings: Bravaco, Chapter 6

Chapter 7

Lesson: Arrays

Topics:

- Creating arrays
- Access array elements

- Array initialization
- The = and == operators
- Passing arrays as arguments
- Returning arrays from methods
- Useful array algorithms
- Two-dimensional arrays
- Command line arguments

Readings: Bravaco, Chapter 7

Chapter 9

Lesson: Classes

Topics:

- Object oriented programming
- Classes and objects
- Objects vs. primitives
- UML
- Commenting code
- Writing code for a class
- Accessor and mutator methods
- Instance methods and attributes

Readings: Bravaco, Chapter 9

Chapter 10

Lesson: More on Classes

Topics:

- The Dice example
- Constructors
- Shadowing
- Overloading constructors
- Static class members
- The this reference
- Garbage collection

Readings: Bravaco, Chapter 10

Chapter 12

Lesson: Inheritance

Topics:

- Superclass constructor
- Overriding methods
- The toString method
- The equals method
- Access modifiers
- The Object class
- The instanceof operator

Readings: Bravaco, Chapter 12